LAURENCE Pierre
BOENS David

Maîtrise informatique
Année 2002-2003

COMPTE-RENDU DU PROJET

Université des Sciences et Technologies de Lille

SOMMAIRE

PRESENTATION GENERALE

ASCQ-PROT est un projet axé bioinformatique. A notre connaissance, nous sommes les seuls de la promotion 2003 de **Maîtrise informatique de l'USTL** à avoir choisi un TER dans ce domaine pourtant très prometteur.

Le projet a été proposé dans le cadre du **Plateau de Protéomique de la Genopôle de Lille**. Nous l'avons donc réalisé en collaboration avec des biologistes spécialisés dans la protéomique. Ils ont défini leur propre « cahier des charges » et effectué les tests d'utilisations en conditions réelles. Nous avons été très libres mais nous avons forcé les rendez-vous réguliers (à peu prés 8) pour permettre et faire suivre l'évolution d'**ASCQ-PROT**. Aussi, nous nous sommes beaucoup documentés pour comprendre toutes ces notions biologiques que nous ne connaissions pas. Cela nous a permis de comprendre rapidement ce qu'on attendait de nous.

La réalisation a été faite en C. **ASCQ-PROT** a pour but d'être utilisé par le Plateau de Protéomique sur des machines Windows. Mais nous avons veillé au fonctionnement de celui-ci sous Linux !

Plus concrètement, **ASCQ-PROT** est une application permettant l'identification de protéines à partir de données expérimentales obtenus par spectrométrie de masse par interrogation des banques de données FASTA. A une liste de masses, on retourne une liste de protéines candidates triées grâce à une fonction de score reflétant la signification ou non de celles-ci.

Les parties suivantes devraient permettre d'éclaircir tout ça.

LES EXPLICATIONS PRELIMINAIRES

Nous allons expliquer quelques points essentiels à la compréhension du rôle du projet **ASCQ-PROT**.

Les protéines

Commençons par des rappels simples mais élémentaires de biologie moléculaire. La cellule est l'élément constitutif fondamental de tout être vivant. On trouve 3000 à 4000 protéines dans une cellule. Une protéine se replie « en pelote », adoptant une configuration spatiale caractéristique de sa fonction.

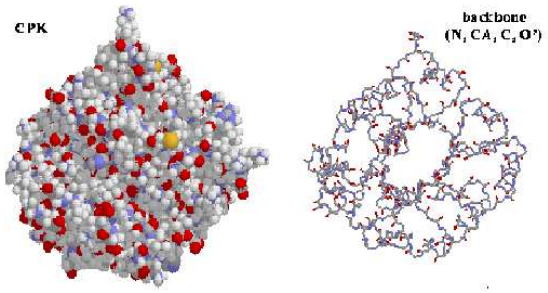


Illustration d'une protéine (triose phosphate isomérase ITIM).

Une protéine est un polymère, une séquence d'éléments fondamentaux, les acides aminés. Il y a 20 acides aminés distincts.

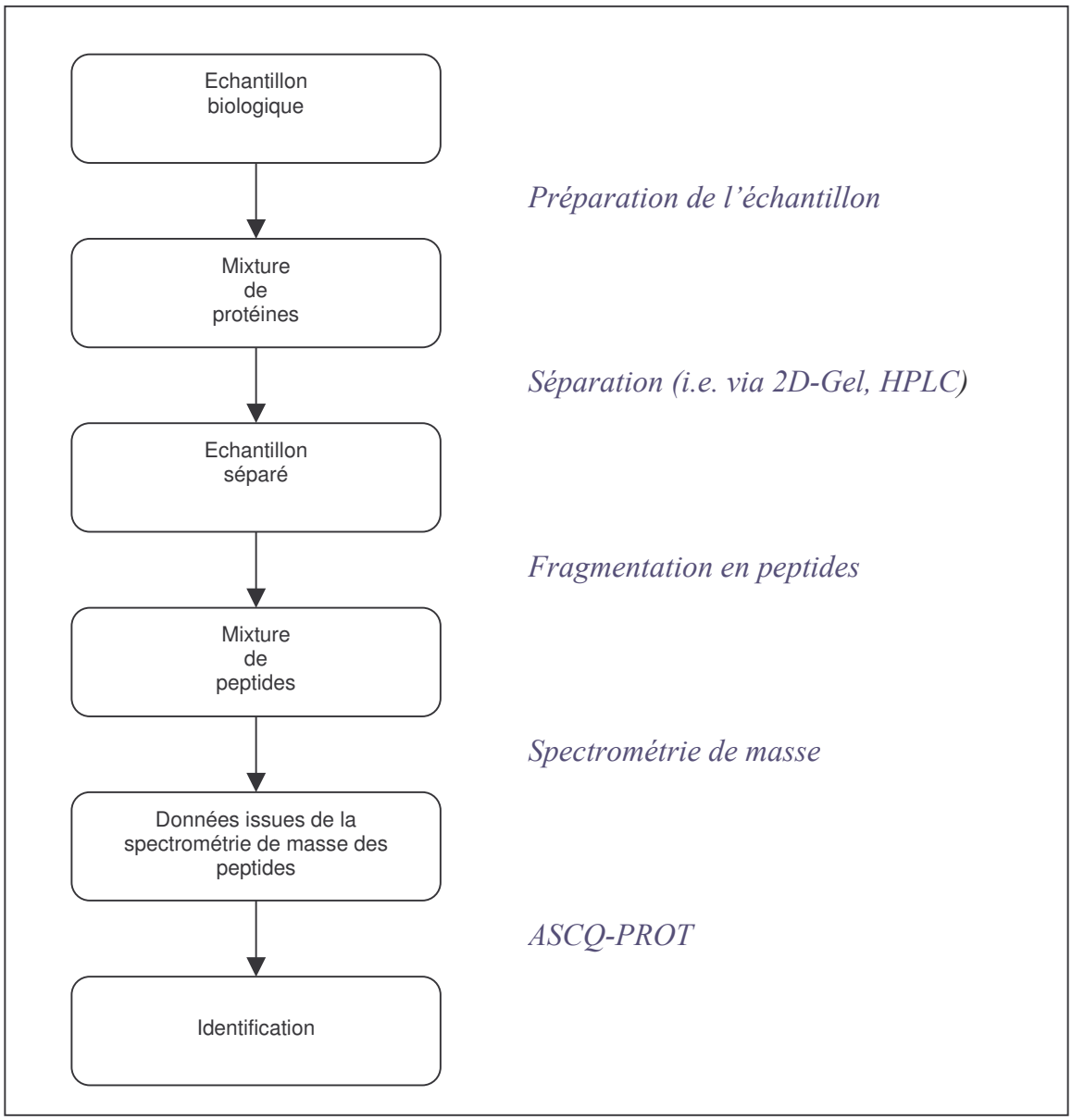
Nom	Symbole	Lettre dans séquence	Masse (en Da)
Alanine	Ala	A	71.03712
Cystéine	Cys	C	103.00919
Asparagine	Asp	D	115.02695
Glutamine	Glu	E	129.04260
Phénylalanine	Phe	F	147.06842
Glycine	Gly	G	57.02147
Histidine	His	H	137.05891
Isoleucine	Ile	I	113.08407
Lysine	Lys	K	128.09497
Leucine	Leu	L	113.08407
Méthionine	Met	M	131.04049
Asparagine	Asn	N	114.04293
Proline	Pro	P	97.05277
Glutamine	Gln	Q	128.05858
Arginine	Arg	R	156.10112
Serine	Ser	S	87.03203
Thréonine	Thr	T	101.04768
Valine	Val	V	99.06842
Tryptophane	Trp	W	186.07932
Tyrosine	Tyr	Y	163.06333

Table des acides aminés.

La longueur d'une protéine est 50 à 1000 acides aminés. Un peptide est un morceau de protéine. De plus, on peut rajouter qu'à chaque acide aminé correspond une masse moléculaire (en Da). On peut donc calculer facilement la masse d'une protéine, c'est la somme (mis à part quelques détails de biochimie…) des masses de ses acides aminés.

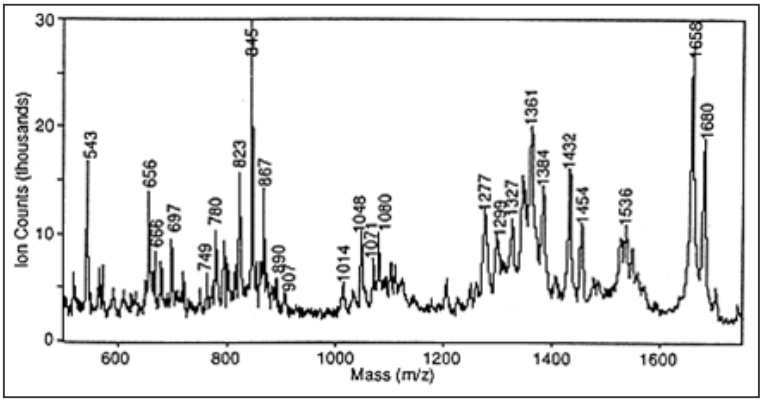
L'identification de protéines

Le schéma ci-dessous résume les différents processus mis en oeuvre afin de récupérer les données en entrée d' **ASCQ-PROT**.



De l'échantillon biologique à ASCQ-PROT.

La méthode la plus utilisée pour l'identification de protéines est basée à partir du « mass fingerprinting » de peptides. La méthode du « mass fingerprinting » entraîne l'utilisation de la spectrométrie de masse pour mesurer la masse des fragments de peptides d'une protéine.



Des pics de masses issus d'une spectrométrie de masse

Une protéine est alors identifiée de manière unique (du moins on l'espère !) en comparant les données expérimentales des masses des peptides avec les valeurs théoriques des séquences de la banque de données. Plus les données expérimentales sont précises et complètes, plus l'identification sera fructueuse.

Masse recherchée	Tolérance de masse (Da)	Nombre de protéines trouvées
1529	1	478
1529.7	0.1	164
1529.73	0.01	25
1529.734	0.001	4
1539.7348	0.0001	2

Effet de la précision de masse

Masses recherchées	Tolérance de masse (Da)	Nombre de protéines trouvées
1529.73	0.1	204
1529.73 1252.70	0.1	7
1529.73 1252.70 1833.88	0.1	1

Effet du nombre de masses

La digestion de protéine

Avant de commencer l'analyse de la spectrométrie de masse, nous avons vu qu'un processus de fragmentation de la protéine est nécessaire. Cela permettra de faciliter l'analyse. Ce processus est généralement fait grâce à une digestion enzymatique. En effet, si on choisit une enzyme dont le fonctionnement est connu, on sait à quels acides aminés celle-ci va couper la protéine. La trypsine est l'enzyme utilisée la plupart du temps (plus de 95%). La trypsine possède la caractéristique de couper la chaîne des acides aminés aux résidus lysine et arginine (aux lettres K et R avec certaines exceptions peu intéressantes à préciser…). Le facteur le plus important est de pouvoir prévoir théoriquement les fragments quand on utilise une enzyme sur les séquences de protéines (d'une banque de données).

Illustrons le processus de digestion enzymatique par un exemple :

<u>Etape 1</u> : on choisit une protéine (le Cytochrome C bovin).

| | |

G D V E K G K K I F V Q K C A Q C H T V E K G G K H K T G P
N L H G L F G R K T G Q A P G F S Y T D A N K N K G I T W G
E E T L M E Y L E N P K K Y I P G T K M I F A G I K K K G E
R E D L I A Y L K K A T N E

<u>Etape 2</u> : on choisit une enzyme (la Trypsine).

<u>Etape 3</u> : on repère les points de clivage (K et R).

| | |

*G D V E **K** G **K** **K** I F V Q **K** C A Q C H T V E **K** G G **K** H **K** T G P*
*N L H G L F G **R** **K** T G Q A P G F S Y T D A N **K** N **K** G I T W G*
*E E T L M E Y L E N P **K** **K** Y I P G T **K** M I F A G I **K** **K** **K** G E*
***R** E D L I A Y L **K** **K** A T N E*

<u>Etape 4</u> : on obtient les 21 fragments suivants.

Position dans la séquence	*Fragment*
1-5	*GDVE**K***
6-7	*G**K***
8-8	***K***
9-13	*IFVQ**K***
14-22	*CAQCHTVE**K***
23-25	*GG**K***
26-27	*H**K***
28-38	*TGPNLHGLFG**R***
39-39	***K***
40-53	*TGQAPGFSYTDAN**K***
54-55	*N**K***
56-72	*GITWGEETLMEYLENP**K***
73-73	***K***
74-79	*YIPGT**K***
80-86	*MIFAGI**K***
87-87	***K***
88-88	***K***
89-91	*GE**R***
92-99	*EDLIAYL**K***
100-100	***K***
101-104	*ATNE*

Les complications

La digestion théorique n'est pas toujours parfaite. L'agent de clivage (l'enzyme choisie) rate parfois certains points de clivage. A la comparaison des masses des fragments du découpage théorique, il faut donc rajouter la comparaison des masses des fragments dont certains clivages sont manqués.

Reprenons l'exemple précédent du Cytochrome C bovin, on avait un petit nombre de fragments (au total : 21 fragments). Supposons un seul clivage manqué et on obtient ces fragments supplémentaires (au total : 41 fragments) :

1-7	GDVE**K**G**K**	54-72	N**K**GITWGEETLMEYLENP**K**
6-8	G**KK**	56-73	GITWGEETLMEYLENP**KK**
8-13	**K**IFVQ**K**	73-79	**K**YIPGT**K**
9-22	IFVQ**K**CAQCHTVE**K**	74-86	YIPGT**K**MIFAGI**K**
14-25	CAQCHTVE**K**GG**K**	80-87	MIFAGI**KK**
23-27	GG**K**H**K**	87-88	**KK**
26-38	H**K**TGPNLHGLFG**R**	88-91	**K**GE**R**
28-39	TGPNLHGLFG**RK**	92-100	EDLIAYL**KK**
39-53	**K**TGQAPGFSYTDAN**K**	89-99	GE**R**EDLIAYL**K**
40-55	TGQAPGFSYTDAN**K**N**K**	100-104	**K**ATNE

Supposons que l'on autorise maintenant deux clivages manqués et on obtient ces nouveaux fragments supplémentaires (au total : 60 fragments) :

1-8	GDVE**K**G**KK**	54-73	N**K**GITWGEETLMEYLENP**KK**
6-13	G**KK**IFVQ**K**	56-79	GIT…EYLENP**KK**YIPGT**K**
8-22	**K**IFVQ**K**CAQCHTVE**K**	73-86	**K**YIPGT**K**MIFAGI**K**
9-25	IFVQ**K**CAQCHTVE**K**GG**K**	74-87	YIPGT**K**MIFAGI**KK**
14-27	CAQCHTVE**K**GG**K**H**K**	80-88	MIFAGI**KKK**
23-38	GG**K**H**K**TGPNLHGLFG**R**	87-91	**KK**GE**R**
26-39	H**K**TGPNLHGLFG**RK**	88-99	**K**GE**R**EDLIAYL**K**
28-53	TGP…LFG**RK**TGQ…DAN**K**	89-100	GE**R**EDLIAYL**KK**
39-55	**K**TGQAPGFSYTDAN**K**N**K**	92-104	EDLIAYL**KK**ATNE
40-72	TG…AN**K**N**K**GIT…EYLENP**K**		

A cette première complication s'ajoute le fait que la masse de certains acides aminés peut être modifiée. Ce phénomène est causé par les modifications post-traductionnelles. Nous avons géré les trois plus fréquentes d'entres elles :

- Modification volontaire pour faciliter l'analyse :
  modification de la cystéine par carbamidométhylation.
- Modification involontaire durant les étapes analytiques :
  oxydation des méthionines.
- Modification biologique :
  phosphorylation des sérines et thréonines.

Concrètement, cela veut dire que les lettres C, M, S et T n'ont pas toujours la même masse. On augmente celle-ci d'une certaine constante. Les modifications ou non peuvent se combiner dans un même peptide. De ce fait, à cause de ces

combinaisons possibles, un peptide peut avoir de nombreuses masses possibles.

Ainsi, si on prend un fragment (14-22) de notre Cytochrome Bovin précédent, on obtient 8 combinaisons, soient 8 masses théoriques possibles.

14-22	CAQCHTVEK
14-22	**C**AQCHTVEK
14-22	CAQ**C**HTVEK
14-22	CAQCH**T**VEK
14-22	**C**AQ**C**HTVEK
14-22	CAQ**C**H**T**VEK
14-22	**C**AQCH**T**VEK
14-22	**C**AQ**C**H**T**VEK

Combinaisons des modifications possibles sur un exemple de fragment.

On peut déjà envisager que la combinaison des deux complications va faire très rapidement augmenter le nombre de masses à comparer (la toute petite protéine Cytochrome C Bovin de notre exemple montre déjà cette explosion). De ce fait, il faut exploiter tous les paramètres de l'utilisateur permettant d'accélérer la recherche :

- Limite de masses non coïncidentes
- Limite de clivage manqué
- Intervalle de masse de recherche dans la banque

Les banques de données

Les grandes banques de séquences généralistes telles que Genbank ou l'EMBL sont des projets internationaux et constituent des leaders dans le domaine. Elles sont maintenant devenues indispensables à la communauté scientifique car elles regroupent des données et des résultats essentiels dont certains ne sont plus reproduits dans la littérature scientifique. Leur principale mission est de rendre publiques les séquences qui ont été déterminées. Un des premiers intérêts de ces banques est la masse de séquences qu'elles contiennent. Pour les tests, nous avons utilisé principalement les trois bases de protéines du tableau ci-dessous.

Nom	Nombre de protéines	Taille
SPROT	120.000	50 Mo
TREMBL	830.000	290 Mo
NCBI	2.000.000	680 Mo

Les banques de protéines que nous avons utilisées.

On peut déjà observer qu'au point de vue informatique cela représente des données imposantes (nombre de protéines - taille des fichiers). Dés le départ, il faut envisager que le nombre d'itérations du programme est le nombre de protéines de la base choisie. De plus, une base comme NBCI représente tout de même 680 Mo, ce qui est encore aujourd'hui à prendre en compte (La mémoire vive d'un ordinateur en 2003 est de 512 Mo).

Le format FASTA

Les banques avec lesquelles nous avons travaillées étaient au format FASTA. Un fichier FASTA est un simple fichier texte. Les séquences sont mises les unes à la suite des autres en respectant un certain format. Une séquence dans le format FASTA commence par une seule ligne de description, suivi par les lignes de données de la séquence. La ligne de description est distinguée des lignes de données grâce au symbole « > » à la première colonne. Il est recommandé que toutes les lignes de texte soient plus petites que 80 caractères.

```
>DKA1_YEAST (P14306) DKA1 protein (NSP1 protein) (TFS1 protein).
MNQAIDFAQASIDSYKKHGILEDVIHDTSFQPSGILAVEYSSS
APVAGNTLPTEKARSKPQFQFTFNKQMQKSVPQANAYVPQ
DDDLFTLVMTDPDAPSKTDHKWSEFCHLVECDLKLLNEATH
ETSGATEFFASEFNKGSNTLIEYMGPAPPKGSGPHRYVFLLY
KQPKGVDSSKFSKIKDRPNWGYGTPATGVGKWAKENNLQL
VASNFFYAETK
```

Un exemple de séquence au format FASTA.

LES OBJECTIFS

Position vis-à-vis des outils «concurrents »

ASCQ-PROT n'est pas en soit une application innovante. En effet, il existe déjà une large variété de logiciels dédiés à l'identification de protéine grâce aux données de la spectrométrie de masse. On peut les classer en trois catégories :

- *Première génération :*
  Ceux qui assignent des scores en fonction du nombre de masses qui ont coïncidé avec une certaine tolérance.
  exemples : PepSea, PepIdent, MultIdent et FragFit.

- *Seconde génération :*
  Ceux qui prennent en compte les clivages manqués, les modifications et dont la fonction de score est un peu plus élaborée.
  exemples : EMowse et MS-Fit.

- *Troisième génération :*
  Ceux qui utilisent un système de score basé sur les probabilités et/ou des techniques de statistiques pour vérifier la validité des résultats.
  exemples : ProFound, Mascot et SEQUEST.

L'objectif premier était de pouvoir au minimum être classé parmi les utilitaires déjà existants. En fait, **ASCQ-PROT** se situe même dans la deuxième catégorie.

Les avantages

Notre application n'est pas une application commerciale de troisième génération. Pourtant, elle possède quelques atouts qui finalement étaient nos objectifs et les raisons de son développement :

- Open source et freeware : nous avons veillé à ne pas coder de façon « tortueuse » pour qu'un biologiste avec un minimum de connaissances en C puisse adapter notre code.
- Utilisable hors ligne : en effet, **ASCQ-PROT** autorise les scripts très facilement et se configure très rapidement contrairement aux sites proposant l'identification de protéine. De plus, pour des questions de rapidité réseau certains paramètres sont bloqués contrairement à **ASCQ-PROT** (clivage manqué illimité par exemple).
- Fonction de score simple et compréhensible : finalement, les utilisateurs reprochent aux outils de seconde génération d'avoir des fonctions de score incompréhensibles et aux outils de troisième génération d'abuser des probabilités. Notre fonction de score donne de bons résultats vis-à-vis de sa simplicité.
- **ASCQ-PROT** propose des informations annexes à chaque protéine candidate comme l'on choisit les utilisateurs potentiels (ici, c'est le logiciel qui s'est adapté aux l'utilisateurs et non pas l'inverse !).

L'UTILISATION

L'installation

L'installation d'**ASCQ-PROT** (Windows ou Linux) est réellement simple. Elle consiste à extraire l'archive appropriée dans le répertoire choisi. **ASCQ-PROT** ne va créer aucun autre fichier que ceux spécifiés par l'utilisateur.

La compilation des sources d'**ASCQ-PROT** a été faite et testée avec **gcc** (Windows et Linux). Grâce au fichier « *Makefile »,* un simple « *make* »dans une console devrait compiler **ASCQ-PROT**. La compilation a aussi été testée avec **Microsoft Developper Studio 1997**. Il suffit d'ouvrir le fichier « *aprot.dsw* ».

Pour d'autres compilateurs ou d'autres systèmes d'exploitations, le code source nécessite peut-être quelques modifications.

Les options

Voici les options d'**ASCQ-PROT** pour se faire une idée de ce que nous gérons :

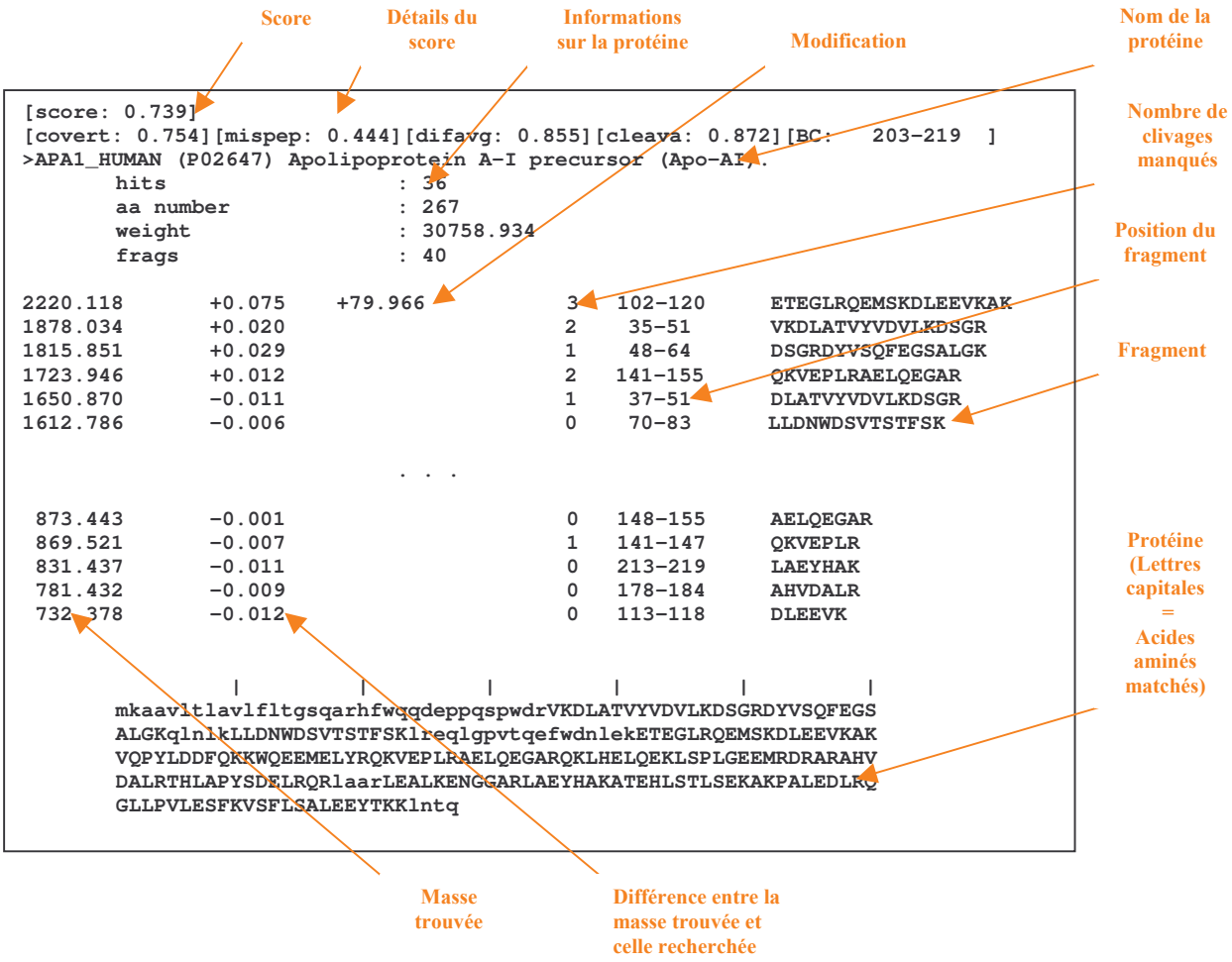
Option	Parametre	Fonction	Commentaires
-d	fichier FASTA	banque de données de protéines à utiliser pour la recherche	-
-w	fichier TEXTE	liste des masses issues de la spectrométrie	-
-o	fichier TEXTE	fichier pour la sauvegarde des résultats de la recherche	un fichier déjà existant est effacé!
-f	fichier FASTA	fichier FASTA contenant les protéines candidates	utile pour faire une recherche en plusieurs passes (sorte d'écrémage progressif…)
-p	fichier TEXTE	fichier de configuration contenant les paramètres de la recherche.	-
-m	nombre	minimum de masse à matcher	par défaut : 5
-c	nombre	Clivages maximum manqués	par défaut : 0
-t	nombre	tolérance pour l'égalité de 2 masses (en Da)	par défaut : 0.2
-s	nombre	score minimum pour être une protéine sélectionnée	>0 et < 1 par défaut : 0.5
-l	nombre	borne inférieure pour la masse des protéines candidates (en Da)	par défaut : 10000
-h	nombre	borne inférieure pour la masse des protéines candidates (en Da)	par défaut : 50000
-modif	-	modifications des acides aminés activées	-
-beep	-	émet un son quand le travail est fini	-
-silent	-	aucun affichage écran	disponible uniquement si option –o…
-help	-	affichage d'une petite aide	-

L'usage

Voici un exemple d'utilisation d'**ASCQ-PROT** (ligne de commande ou script) :

aprot -d .\fasta\sprot.fas -w .\test\caro_apo137 -m 10 -t 0.1 -c 5 -t 0.1 -s 0.4 –o ./res/res_apo137.txt

En résultat, on obtient un fichier « res_apo137.txt » qui contient les résultats des protéines candidates à la recherche dans l'ordre de leur score. Un résultat de protéine se présente ainsi :



Pour être utilisé, **ASCQ-PROT** nécessite une banque de donnée. (La banque SwissProt peut être téléchargée sur http://us.expasy.org).

ASCQ-PROT possède une documentation html en anglais (le site dédié en fait) dans le répertoire « ./doc ». Elle est aussi complète que ce rapport.

A la racine du projet, nous avons mis des scripts de tests (les fichiers test.*.bat) et dans le répertoire « ./cfg » des exemples de fichiers de configuration. Cela permet à l'utilisateur de voir le fonctionnement d'**ASCQ-PROT** très rapidement en s'affranchissant de lire la documentation (en éditant ces fichiers pour son propre usage).

La fonction de score

Pour l'instant, notre fonction de score est très simple et relativement efficace (d'après les utilisateurs du Plateau de Protéomique, **ASCQ-PROT** conduit à de meilleurs résultats que les logiciels commerciaux dont ils disposent !).

SCORE =

1/2 x (taux de couverture de la masse de la protéine)

+

1/6 x (proportion de masses trouvées par rapport au nombre de masses à trouver)

+

1/6 x (différence moyenne / tolérance)

+

1/6 x (nombre moyen de clivages ratés / clivages manqués autorisés)

LA REALISATION

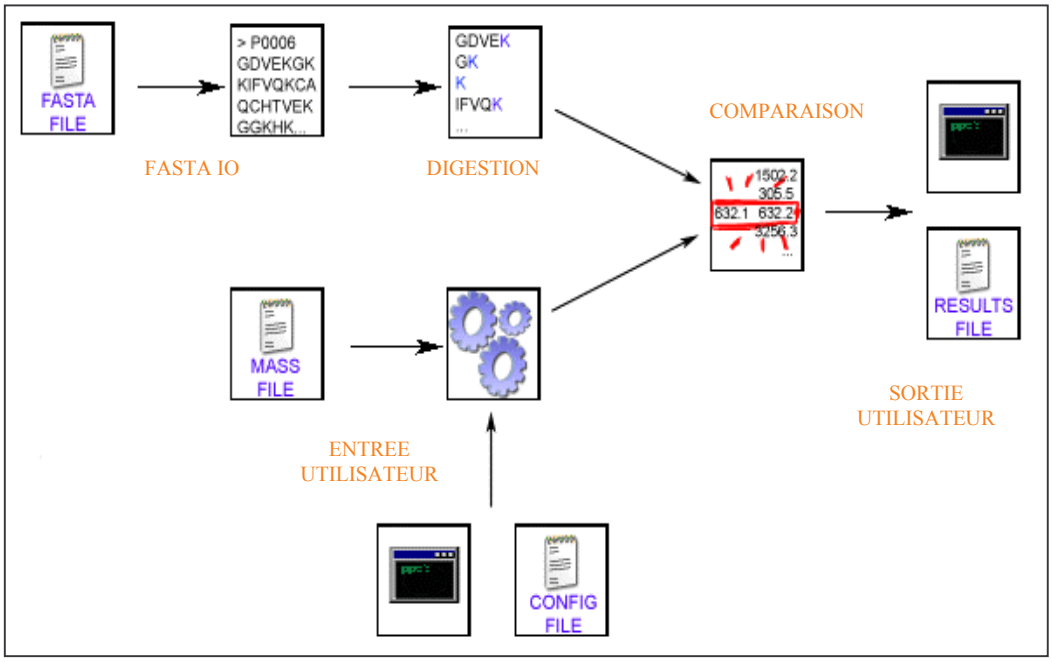
Le principe

Pour tout (protéine de la banque de données)
    digérer ( )
    comparer ( )
    Si ((paramètres utilisateur) respectés)
        ajouter-liste-resultat ( )

digérer :
construction de la liste de peptides par digestion enzymatique.

comparer :
comparaison des masses recherchées avec celles obtenues par la digestion (prise en compte des modifications et des clivages manqués).

La conception générale



Le fonctionnement général d'ASCQ-PROT.

Le code source est structuré et divisé en 5 parties :

- *main.c*, *tools.h*, et *tools.c* (et aussi *error.h* et *error.c*) :
  Cette partie permet de lancer l'exécutable et de récupérer les paramètres de l'utilisateur (parse les paramètres de la ligne de commande et parse les fichiers de configuration). Il y a aussi quelques outils (comme la lecture du fichier de la table des acides aminés ou la gestion des erreurs).
- *seqio.h*, *seqio.c* :
  Cette partie gère les entrées-sorties des fichiers FASTA.
- *weightList.h*, *weightList.c* :
  une implémentation d'une liste dynamique de nombres doubles.
- *cleavage.h*, *cleavage.c* :
  Cette partie gère la description d'un agent de clivage (grâce à une sorte d'expression régulière).
- *digest.h*, *digest.c* :
  Le noyau d'ASCQ-PROT : la digestion de protéine, la comparaison des masses et le calcul de score se trouvent dans cette partie.

Les pistes pour la maintenance

Le but de cette partie (ainsi que les commentaires du code source) est d'accélérer une éventuelle maintenance d'**ASCQ-PROT**.

- *main.c*, *tools.h*, et *tools.c* (et aussi *error.h* et *error.c*) :
  Le code source de *main.c* est long mais relativement simple.
  La principale fonction est ... la fonction ***main ()***.
  Tout d'abord, elle initialise quelques variables :
  aaTable : la structure qui contient les masses des acides aminés,
  hitTable : la structure dynamique qui contient tous les hits,
  param : la structure qui contient tous les paramètres utilisateur (paramètres par défaut) …
  Ensuite, elle récupère tous les paramètres utilisateur (boucle "***while (arg(c))*** ") et met à jour la variable ***param***.
  Quand le parsing est fini, on contrôle la validité des paramètres…
  La fonction ***error ()*** s'occupe de la gestion des erreurs (dans *error.c*).
  Ensuite, le main lit chaque séquence de protéine de la banque de données (boucle "***while (nextEntry (&Offset))***") grâce à la fonction ***nextEntry ()*** (dans *seqio.c*), la digère avec la fonction ***digest*** et confronte les masses avec la fonction ***compare*** (fonctions dans *digest.c*). Si la protéine correspond à tous les paramètres de l'utilisateur, elle est ajoutée à ***hitTable***.

- *seqio.h*, *seqio.c* :
  La fonction ***nextEntry (long *offset)*** est la fonction essentielle de *seqio.c* .
  Elle permet d'obtenir l'entrée suivante d'un fichier FASTA ouvert grâce à la fonction ***fastafopen ()***. Elle met à jour 3 choses : le buffer prévu pour contenir la séquence FASTA courante (***SeqBuffer***), le buffer prévu pour contenir sa description (***DescBuffer***), et l'offset de l'entrée dans le fichier FASTA

(paramètre ***offset***).
Cet offset peut être stocké dans le but de pouvoir relire une entrée FASTA directement dans le fichier grâce à la fonction ***getEntry (long *offset)***.
Les deux précédents buffers peuvent être très facilement lus grâce aux fonctions ***getSeq ()*** et ***getDesc ()*** qui renvoient des chaînes de caractères.
Attention, ***SeqBuffer*** est rempli avec des caractéres minuscules !

- *weightList.h*, *weightList.c* :
  Ces deux fichiers implémentent juste une liste dynamique de doubles avec des fonctions outils associées. Pour créer une telle liste, il suffit juste d'utiliser la fonction ***wl_malloc ()***. La liste peut être effacée de la mémoire grâce à la fonction ***wl_free ()***. La taille de la liste est automatiquement réallouée ... donc le nombre d'éléments est illimité (sauf par la taille de la mémoire !)
  Il y a aussi une implementation du quicksort (fonction ***wl_quicksort ()***) pour trier une telle liste (dans l'ordre décroissant).

- *cleavage.h*, *cleavage.c* :
  Nous avons défini un point de clivage ***cp_t*** comme un nouveau type. Nous avons une fonction (***stringToCpl(char *s)***) qui convertit une expression en une liste (type ***cpl_t***) de ***cp_t*** . L'expression doit respecté la grammaire BNF suivante :
  letter ::= ***a*** | ***b*** | ***c*** | ... | ***z***
  letter_list ::= letter { letter }
  cp ::= letter | letter***+***letter_list | letter***-***letter_list | letter***+***letter_list***-***letter_list
  cpl ::= cp { ***;*** cp }
  letter est le point de clivage.
  +letter_list est la liste des lettres après letter qui font que letter n'est pas un point de clivage.
  -letter_list est la liste des lettres avant letter qui font que letter n'est pas un point de clivage.

  Exemple : pour la Trypsine, l'expression est « k+kp-k;r+f » et signifie « clive sur le k si il n'est pas suivi par k ou p et si il n'est pas précédé par k et clive sur le r si il n'est pas suivi par f »

  Bien qu'**ASCQ-PROT** n'utilise que la Trypsine pour l'instant, cette partie permettra de pouvoir définir d'autres agent de clivages par la suite.

- *digest.h*, *digest.c* :
  La fonction ***digest ()*** coupe une séquence d'acides aminés en fragments (voir type ***frag*** dans *digest.h*) et met à jour la variable globale ***Frag*** avec eux.
  Le processus de digestion utilise les outils définis dans *cleavage.h*. Le cœur d'ASCQ-PROT est dans *digest.c* : c'est la fonction ***compare***. Elle compare les paramètres de masses de l'utilisateur avec les masses des fragments de ***Frag***.
  Nous avons fait en sorte d'accélérer cette partie. Par exemple, ne pas effectuer les calculs inutiles lorsqu'on sait qu'au bout d'un certain nombre de clivages manqués on dépasse déjà la masse recherchée…
  La rapidité peut aussi être augmentée grâce au système de recherche en plusieurs passes (laissées à la charge de l'utilisateur pour des raisons de recherche réellement significative).

CONCLUSION

Nous avons atteint nos objectifs de réaliser un outil satisfaisant les demandes des utilisateurs. Certaines options semblent même être très appréciées. Par exemple, on peut citer l'affichage des masses qui n'ont pas matché pour pouvoir relancer une autre recherche par la suite (jusque là fait à la main !). Nous avons aussi apprécié le contact auprès des biochimistes. C'est intéressant d'écouter les demandes de personnes non initiées au domaine de la programmation. En effet, parfois une demande à leurs yeux très simple, peut devenir un véritable casse-tête pour nous. Et inversement, parfois c'est une demande qui est très importante pour eux et qui dans notre code ne prendra pas plus d'une ou deux lignes !

De plus, nous tenons à dire que nous prolongeons ce projet. Tout d'abord, nous avons créé un site dédié à **ASCQ-PROT** et qui va être hébergé sur le site du Plateau de Protéomique de la Genopôle deLille. Nous restons donc à l'écoute de toutes les suggestions des utilisateurs. Nous sommes aussi embauchés en CDD au sein du Plateau de Protéomique, pour ces vacances, pour le développement d'un programme « scientifiquement innovant » d'identification de protéine (à partir d'**ASCQ-PROT**).

BIBLIOGRAPHIE

- http://www.infobiogen.fr
- http://www.matrixscience.com
- http://us.expasy.org
- http://www.unimod.org
- The Protein Identification Problem
  School of Computer Science (McGill University)
  (2002) A. Guérette, B. Carrillo et W. Wu
- Peptide Sequencing Using Mass Spectrometry (Phase II Report)
  (2001) C. Zhuo, W. Xinyu, Z. Wenbin et Y. Xiang
- Mass Spectrometry for the study of Protein - Protein Interactions
  (2001) D. Figeys, L. D. McBroom, et M. F. Moran
- Algorithmic complexity of protein identification : Combinatorics of weighted strings
  (2001) M. Cieliebak, T. Erlebach, Z. Lipták, J. Stoye, E. Welzl
- PepIdent2 : Identification des protéines par "mass fingerprinting"
  (2000) SwissProt
- Emboss MOWSE documentation
  (2000) A. Bleasby
- Comment tout savoir sur la biologie moléculaire en moins de 90 minutes ?
  (2003) H. Touzet

REMERCIEMENTS

Nous tenons à remercier toutes les personnes qui ont montré un intérêt, qui ont passé du temps avec nous, qui ont testé, et qui ont fait évoluer **ASCQ-PROT** par leurs idées et leurs suggestions.

Particulièrement à :

Christian ROLANDO,
Directeur du Plateau de protéomique de la Génopole de Lille.
Auteur du sujet du projet et qui a été notre véritable guide pendant le développement d'**ASCQ-PROT**.

Cécile CREN
Pour ses explications de protéomique, pour ses suggestions et ses tests.

Caroline et Florence
Pour leurs exemples qui nous ont permis de tester **ASCQ-PROT**.